

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :

H04L 12/24

A1

(11) International Publication Number:

WO 96/31035

(43) International Publication Date:

3 October 1996 (03.10.96)

(21) International Application Number: PCT/US96/04332

(22) International Filing Date: 29 March 1996 (29.03.96)

(30) Priority Data:

08/412,955

29 March 1995 (29.03.95)

US

08/558,425

16 November 1995 (16.11.95)

US

(71) Applicant: CABLETRON SYSTEMS, INC. [US/US]; 35 Industrial Way, Rochester, NH 03867 (US).

(72) Inventors: POLIQUIN, Lynn, R.; 190 Mountain Road, Goffstown, NH 03054 (US). ARROWSMITH, Russell; 8 Arbor Street, Merrimack, NH 03054 (US). LEWIS, Lundy; 480 Greenville Road, Mason, NH 03048 (US). TRACY, William; 60 Sleight Road, Chelmsford, MA 01824 (US).

(74) Agent: HENDRICKS, Therese, A.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

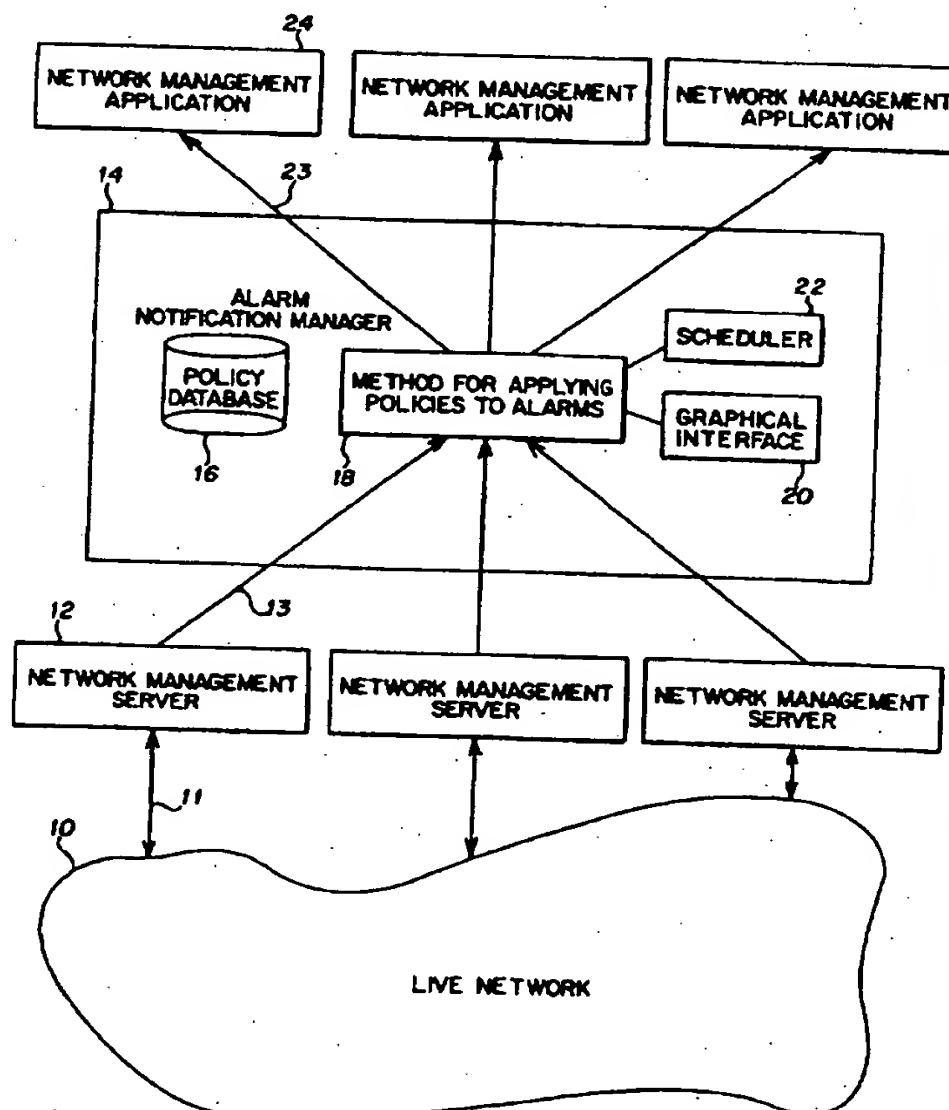
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR POLICY-BASED ALARM NOTIFICATION IN A DISTRIBUTED NETWORK MANAGEMENT ENVIRONMENT

(57) Abstract

Apparatus and method for receiving alarms from multiple network management servers (12) and applying a plurality of policy-based filters (18) to the alarms. The filters may be named and stored in a database (16), and application of the policy-based filters may be scheduled for different times. The same policy-based filters may be applied to one or more multiple network management applications (24). The invention allows greater control over which alarms get reported to network management applications and provides a means to ensure consistency of reported alarms across multiple network management applications. A telephonic alarm notification method and apparatus incorporates the policy-based filters and the capability to process alarms from multiple network segment servers so that users can be accurately notified of critical alarms generated in large and complex communications networks, via a public communications system.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

**METHOD AND APPARATUS FOR POLICY-BASED
ALARM NOTIFICATION IN A DISTRIBUTED
NETWORK MANAGEMENT ENVIRONMENT**

5

Related Cases

This is a continuation-in-part of copending and commonly owned U.S. Serial No. 08/412,955 filed March 29, 1995 by Arrowsmith et al. entitled "METHOD AND APPARATUS FOR POLICY-BASED ALARM NOTIFICATION IN A DISTRIBUTED NETWORK
10 MANAGEMENT ENVIRONMENT."

Field of the Invention

The present invention relates to alarm notification in a communications network and more specifically to a method and apparatus for receiving alarms from multiple network
15 management servers, applying policies to those alarms and forwarding the alarms that conform to the policies to one or more network management applications, such as a telephonic alarm notification method and apparatus.

Background of the Invention

20 Spectrum™ is a model-based network management system, sold by Cabletron Systems, Inc., Rochester, New Hampshire, for maintaining and processing information pertaining to the condition of a communications network and providing the same to a user. For example, Spectrum™ will periodically poll a network device to request information, such as the number of packets sent on the network in a given time and the number of errors that
25 occurred. If the error rate is above a predetermined limit, an error alarm is logged in the Spectrum™ database, an alarm sent to the user interface to notify the network manager, and a message is sent to shut off the corresponding network device.

Alternatively, if no response was received from the network device when it was polled, the reason for the loss of contact should be determined so that appropriate action, such
30 as a service call, can be taken. In a network environment, loss of contact with a network device may be due to failure of that network device or to failure of another network device that is involved in the transmission of a message.

In many prior art network management systems, the network administrator was typically provided with a list of possible causes of a fault and was required to isolate the fault

based on his experience and knowledge of the network. In Spectrum™, the system itself isolates network defaults using a technique known as Status Suppression. Spectrum™ maintains a database of models for each network device. When contact between a model and its corresponding network device is lost, the model sets a fault status and initiates the fault isolation technique. The model (first model) which lost contact with its corresponding network device (first network device) determines whether adjacent models have lost contact with their corresponding network devices; adjacent network devices are defined as those which are directly connected to a specified network device. If adjacent models cannot contact the corresponding network devices, then the first network device cannot be the cause of the fault, and its fault status in the first model will be overridden. By suppressing the fault status of the network devices which are determined not to be defective, the defective network device can be identified. Once the fault has been isolated, the condition of the defective device can be updated in the Spectrum™ database, a control message can be sent shutting off the defective device, and the network administrator can be notified via the user interface.

Spectrum™'s associated SpectroGRAPH™ user interface provides a graphical view into the network models. An alarm log view 123, shown in Fig. 1, includes an area 120 for the listing of current alarms, and an area 122 for displaying information pertaining to a selected alarm. The user may click on a particular alarm in the listing of current alarms to obtain more information. A multi-function icon 124 representing the network device having a fault is displayed in area 122, with one or more text fields 126 and 128 which provide information to the user regarding the cause of the alarm and the status of the device. By clicking on specified areas of the icon 124, the user can obtain further information regarding the device for which an alarm is registered.

Another method for fault management in large communications networks is to use a so-called "trouble-ticketing" system. This system provides a number of tools that can be used by network users, administrators, and repair and maintenance personnel. The basic data structure, a "trouble-ticket", has a number of fields in which a user can enter data describing the parameters of an observed network fault. A trouble-ticket filled out by a user may then be transmitted by, for example, an electronic mail system to maintenance and repair personnel. A trouble-ticket describing a current network fault that needs to be acted on is called "an outstanding trouble-ticket". When the network fault has been corrected, the solution to the problem, typically called a "resolution" is entered into an appropriate data field

in the trouble-ticket and the trouble-ticket is said to be completed. The system provides for storage of completed trouble-tickets in memory and thus a library of such tickets is created, allowing users, administrators, and maintenance and repair personnel to refer to the stored completed trouble-tickets for assistance in determining solutions to future network faults. An
5 example of a trouble-ticketing system is the ACTION REQUEST system, developed by Remedy Corporation, Mountain View, California, and sold by Cabletron Systems, Inc., Rochester, New Hampshire.

ARS Gateway™ is a network management application sold by Cabletron Systems, Inc. which receives fault information from the Spectrum™ system and
10 automatically generates a trouble-ticket that may be processed by the ACTION REQUEST system. This system is further described in copending and commonly owned U.S. Serial No. 08/023,972 filed February 26, 1993 by Lundy Lewis, and entitled "Method and Apparatus For Resolving Faults In Communications Networks," and which is hereby incorporated by reference in its entirety.

15 The Spectrum™ system is described in U.S. Patent No. 5,261,044 issued November 9, 1993 to Roger Dev et al., which is hereby incorporated by reference in its entirety. The Spectrum™ network management system is commercially available and also described in various user manuals and literature available from Cabletron Systems, Inc., Rochester, New Hampshire.

20 Other network management platforms and applications for the basic filtering of alarms which are commercially available include: (1) HP OpenView, Hewlett Packard Corp., 3000 Hanover Street, Palo, CA 94304; (2) LattisNet, SynOptics Communications, 4401 Great American Pkwy., Santa Clara, CA 95054; (3) IBM Netview/6000, IBM Corp., Old Orchard Road, Armonk, NY 10504; and (4) SunNet Manager, SunConnect, 2550 Garcia
25 Ave, Mountain View, CA 94043.

Unfortunately, in the prior art systems alarms can only be received from one network management server. Also there is no provision for applying the same policy-based filter to multiple network management applications.

Thus, it is an object of the present invention to provide greater control over
30 which alarms get reported to network management applications and to provide a means to ensure consistency of reported alarms across multiple network management applications.

An example of a network management application that uses reported alarms is SpectroPHONE™, sold by Cabletron Systems, Inc., Rochester, New Hampshire. SpectroPHONE™ provides remote access to Spectrum™ alarm information from any Touch-Tone phone. SpectroPHONE™ allows the user to make remote queries via the public
5 telephone communications system and can be set to automatically notify the user of alarm conditions.

Fig. 15 shows a typical SpectroPHONE™ configuration. Alarms 150 are generated in a communications network, and a virtual network manager (VNM) 152 manages the information regarding those alarms. SpectroPHONE™ uses a Computerfone unit 154 as
10 an intermediary between a telephone 156 or a pager 158 (or any other device on the public telephone communications system) and the alarm information from the VNM 152. The Computerfone unit 154 also interprets input from a remote Touch-Tone keypad and then produces voice output from the alarm information for the listener at the remote telephone 156 or pager 158.

15 SpectroPHONE™, version 3.0, is a prior art telephone notification method and apparatus that collects and reports alarm information for small communications networks or for isolated segments of such networks via a network management platform such as Cabletron System's Spectrum™. Further descriptions of the prior art version of SpectroPHONE™ are available from Cabletron Systems, Inc., Rochester, New Hampshire.

20 Today's networks are much larger and more complex than the networks of the past. As a result, the network management platforms often logically divide them into segments for performance and diagnostic assessments. Since the prior art version of SpectroPHONE™ can monitor only a single segment of the network at a given time, an instance of the method and apparatus must be installed on each segment of the network for
25 complete monitoring of the network.

This installation requirement can result in greater acquisition and maintenance costs, inconsistent data collection, missed notifications, and an inability to visualize network segment failures in relation to the whole network. Moreover, user intervention is required if information is required from different segments. The user must change the resource files
30 which tell the Network Management Platform which segment to query. The user then must stop and restart SpectroPHONE™. User error could result in invalid resource information,

- 5 -

and new segments may not be visible to SpectroPHONE™, version 3.0. All of these situations would result in an interruption in service.

SpectroPHONE™, version 3.0, polls the network periodically for alarm information. In a large communications network, hundreds of alarms can arise, but typically only a small number of them are critical enough to warrant immediate attention. An example of a failure requiring immediate attention is a power outage on a central device connecting many other devices. Thus, filtering of a large number of alarms is performed in the prior art telephonic method and apparatus also so that the user is notified of the alarms that are critical to system performance. However, the prior art telephonic notification method and apparatus contain limited filtering capabilities, based only on the name and type of device on the network, and on the severity and type of failure.

Thus, it is a further object of the present invention to incorporate the system alarm notification manager of the present invention into a new SpectroPHONE™ application. As a result, communications network administrators can be notified over the public telephone communications system regarding failures on a large and complex communications network with accuracy and regarding only failures that are critical for maintaining the performance of the network.

Summary of the Invention

The present invention is directed to an apparatus and method of alarm notification which includes: a) receiving alarms from multiple network management servers; b) assigning policy-based filters to associated network management applications; and c) applying the assigned policy-based filters to the alarms and for the alarms that pass the filters, generating an alarm notification forwarding the same to the associated network management applications.

In an embodiment described herein, a user designates a plurality of such filters, which constitute an alarm notification policy, to one or more associated network management applications. The policy-based filters are stored in a database, and a tag is assigned for identifying each filter. The same filters may be assigned to multiple applications.

In a further embodiment, the user may schedule the assignment of such policy-based filters to occur at a designated time in the future. For example, a user may pick a policy from a list of available policies to associate with a selected application, and then

designate the frequency with which the policy is applied, e.g., once, hourly, daily, weekly or monthly.

Furthermore, the invention can be used in the same mode as similar tools in the prior art, i.e., with one alarm-forwarding component for each network management system/
5 network management application pair, or alternatively as a single entity in a distributed network management environment.

In another embodiment, the apparatus and method of alarm notification of the prior mentioned embodiments is incorporated into a telephonic alarm notification method and apparatus. In this embodiment, a communications network administrator is notified of alarms
10 that may have been generated on multiple segments of a communications network and that passed predetermined policy-based filters, via a public telephone communications system.

These and other features of the present invention will be more fully described in the following detailed description and figures.

Brief Description of the Drawings

15 Fig. 1 is an example of an alarm log display provided by the prior art Spectrum™ network management system.

Fig. 2 is a block diagram of an alarm notification manager in accordance with the present invention, in use with multiple network management servers and multiple network
20 management applications.

Fig. 3 is a flow chart illustrating the application of policy-based filters to an alarm, and forwarding of the alarm which passes the filters to an application in accordance with this invention.

Fig. 4 is an example of an Associations window display of the alarm
25 notification manager.

Fig. 5 is an example of a New Association window display of the alarm notification manager.

Fig. 6 is an example of a Modified Association window display for the alarm notification manager.

30 Fig. 7 is an example of a Scheduler window display for the alarm notification manager.

Fig. 8 is an example of a Policies window display for the alarm notification manager.

Fig. 9 is an example of an Open Policy window display for the alarm notification manager.

5 Fig. 10 is an example of an Add Filter Values window display for the alarm notification manager.

Fig. 11 is an example of an Alarm Age window display for the alarm notification manager.

10 Fig. 12 is an example of a New Policy window display for the alarm notification manager.

Fig. 13 is a block diagram illustrating two separate processes between the network management application and the alarm notification manager.

Fig. 14 is a block diagram illustrating a central processing unit and memory for use in this invention.

15 Fig. 15 is a typical configuration of a communications system using a prior SpectroPHONE™ application.

Fig. 16 is a block diagram illustrating the incorporation of policy-based filtering into the telephonic alarm notification method and apparatus of the present invention.

20 Fig. 17 is a block diagram illustrating the incorporation of the System Alarm Notification Method (SANM) of the present invention into a new SpectroPHONE™ application.

Fig. 18 is an example of a Graphical User Interface (GUI) window display for a new SpectroPHONE™ application.

25 Detailed Description

The present invention is directed to an alarm notification manager which receives alarms from multiple network management servers, allows an unlimited number of filters to be defined within one policy, allows policies to be named and stored in a database, allows policies to be scheduled for different times, and allows the same policy to be applied
30 to one or more network management applications.

As illustrated in Fig. 2, a live network 10 is connected by links 11 to one or more network management servers 12 which monitor the network. The servers detect errors

or faults on the network and send alarm information to the alarm notification manager 14 via links 13. The alarm notification manager includes a policy database 16, method for applying policies to alarms 18, graphical interface 20, and scheduler 22. The manager applies policy-based filters to the alarm messages received from the servers, and for those alarms
5 which pass the filter criteria, an alarm message is sent to the appropriate network management application 24 via links 23.

In a specific embodiment described herein, a plurality of distributed SpectroServers™, part of the Spectrum™ system sold by Cabletron Systems, Inc., Rochester, New Hampshire, are used to model the live network 10, and several Spectrum™ applications
10 receive the filtered alarm messages from the manager 14. These components have been implemented in the object-oriented programming language C++. However, the invention is not tied to any particular language nor to any particular products used in network management.

15 The Spectrum™ Network Management System

An understanding of the present invention is furthered by an understanding of the model-based network management system known as Spectrum™, which is described in U.S. Patent No. 5,261,044, issued November 9, 1993 to R.Dev et al., and hereby incorporated by reference in its entirety. The Spectrum™ network management system is commercially
20 available and also described in various user manuals and literature available from Cabletron Systems, Inc., Rochester, New Hampshire.

In summary, Spectrum™ is a system for maintaining and processing information pertaining to the condition of the computer network and providing the same to a user, the network including a plurality of network entities such as computer devices and
25 software applications being executed on such devices. The system includes a virtual network machine, comprising a programmed digital computer, wherein a program is implemented using an object-oriented programming language such as C++, Eiffel, SmallTalk, and Ada. The virtual network consists of interrelated intelligent models of network entities and relations between network entities, including means for acquiring network data pertaining to
30 the condition of a network entity from the corresponding network entity. The virtual network further includes means for maintaining objects which include network data relating to the corresponding network entity and one or more inference handlers for processing the network

data, the inference handlers being responsive to changes occurring in the same and/or a different object. The network data can then be transferred to a user interface coupled to the virtual network machine, for supplying the network data to a user.

Thus, the models are implemented as software "objects" containing both "data" (attributes) relating to the corresponding network entity and one or more "inference handlers" (functions) for processing the data. See Grady Booch, "Object-Oriented Analysis And Design, With Applications," 2nd Edition, Benjamin/Cummings Publishing Co., Redwood City, CA, Chapter 2, 1994. The inference handlers are initiated by predetermined virtual network events, such as a change in specified network data in the same model, a change in specified network data in a different model, and predefined events or changes in models or model relations. Information pertaining to the condition of the network entity can be obtained from the network entity by polling the same, can be automatically received from the network entity (without polling), or can be inferred from data contained in other models. An alarm condition may be generated when the network data meets a predetermined criteria. Events, alarms and statistical information from the virtual network are stored in a database and are selectively displayed for the user.

The data in the Spectrum™ database may be used for generating topological displays of the network, showing hierarchial relationships between network devices, isolating a network fault, and reviewing statistical information.

Spectrum™ allows for collective management of autonomous local area networks (LANs), with equipment from different vendors. It complies with the current Simple Network Management Protocol (SNMP) standards, and can also accommodate other standard and proprietary protocols. The virtual network machine preprocesses the raw information coming from the network devices in order to construct a model of the network's current status and performance characteristics. Network elements that cannot be directly communicated with (e.g., cables and buildings) can infer their status from the status of the devices connected to (or contained within) them. The virtual network machine provides a consistent interface for management applications to access any of the information in the model and thereby provides these applications with a unified view of the network.

Spectrum™'s associated SpectroGRAPH™ user interface provides a highly graphical multi-perspective view into the network model. SpectroGRAPH™ enables the user to navigate through a landscape in which cables, networks, local area networks and even

- 10 -

rooms show up as icons, and which icons indicate the health and performance characteristics of those elements. These icons can be further queried for additional information.

SpectroGRAPH™'s main function is to visually present to the user the model within the virtual network machine. It allows the user to navigate freely within the network model, only limited by the access rights assigned by the network administrator. The information can be accessed at varying degrees of detail, from a macro overview, to the devices and cables which connect them. In addition to its navigation functions, SpectroGRAPH™ provides an alarm management facility, an event log window, a reporting facility, a find facility, and other features.

The above description of the Spectrum™ system provides a context for an understanding of the present invention.

The Alarm Notification Manager

The following definitions are helpful to an understanding of the present invention:

SANM SPECTRUM™

Alarm Notification Manager

Policy

A set of criteria which a given alarm must satisfy in order to be passed to the application with which the policy is associated. A policy may consist of one or more filters.

Filter

A set of filter parameters and associated filter values. Multiple filters define multiple sets of values for the filter parameters.

Filter Parameter

A data type such as model name or IP subnet for which the user can specify a value or list of values. SANM provides the user with a fixed list of filter parameters.

Association

When the user associates a policy with an application, he is specifying the filter criteria that SANM should apply to the alarms it sends to the application.

- 11 -

A filter consists of a list of filter parameters and a list of associated filter values. A user (of a network management application) specifies the value(s) that each filter parameter can take in order for a given alarm to pass the filter criteria. The following is a list of representative filter parameters:

- 5 • model name
- model type name
- device IP subnet
- device location
- alarm severity
- 10 • alarm age
- SpectroSERVER host name
- landscape name
- alarm cause

15 The value for each of the above filter parameters would be received from Spectrum™, except for the alarm age parameter. The alarm age parameter is used internally by SANM and specifies the length of time that it should hold an alarm before sending it to an application. If the alarm is cleared by Spectrum™ during this time, it is not sent to the application. This feature may be used to filter out transient alarms.

20 Each filter value also has a corresponding flag which indicates whether it should be negated. For example, if the negate flag is set for a model type name value of Hub_CSI_IRM3, this filter value states that all alarms for models NOT of type Hub_CSI_IRM3 should pass.

25 More complex filtering can be achieved by defining multiple filters within a policy. Each filter specifies a separate set of filter values.

SANM performs a logical AND of all the filter criteria within a filter and performs a logical OR between all filters within a policy.

For example, a policy contains two filters as follows:

Filter 1

30 Model Type: Rtr_Cisco

Landscape: wiz

- 12 -

Filter 2

Model Type: Rtr_Wellfleet

Landscape: brat

SANM would apply this policy to a given alarm as follows:

IF the alarm has:

model type Rtr_Cisco AND landscape wiz

OR

model type Rtr_Wellfleet AND landscape brat

THEN send the alarm to the application.

Each filter also contains a filter tag, which is a text string that the user enters.

This tag, which is included in the alarm notification, identifies which filter(s) passed and can be used by an application to perform routing of alarms.

For example, a different user name can be entered in the filter tag field of each filter, so that if the criteria in one filter pass, the application will notify a particular user, whereas if the criteria in another filter pass, the application will notify a different user. If multiple filters pass, a list of corresponding filter tags is sent in the alarm notification.

Another example of the SANM filtering mechanism is shown in Fig. 3. In this figure, the criteria listed within each filter are the criteria for which values have been specified by the user. It can be seen from this example that all filters are applied in parallel to a given alarm (i.e., a logical OR is performed between filters). However, all criteria within a given filter must be satisfied for the alarm to pass the filter (i.e., a logical AND is performed between the criteria within a given filter). Since, in this example, the alarm passes the criteria in filters 1 and 3, an alarm notification containing filter tags "A" and "C" is sent to the application.

Policies and the associations between policies and applications are stored in the SPECTRUM™ database. This means that the same policies are available to any client machine running SANM. It also means that the policy names contained in event messages logged by SANM have significance to all client machines using the same SPECTRUM™ database.

- 13 -

1.0 Alarm Notification

After an application has registered with SANM to receive alarms, an alarm notification is sent to that application each time an alarm is received from SPECTRUM™ that passes the criteria specified in the policy associated with that application. The

5 information contained in each alarm notification consists of the real-time values of each filter parameter, plus the values of the following parameters:

- model handle
- model type handle
- model condition value
- 10 • model security string
- alarm ID
- alarm time
- alarm probable cause
- alarm status
- 15 • event message associated with alarm
- assigned repair person
- user-clearable flag

One exception to this is that an IP subnet address may be specified as a filter
20 criterion, but the full IP address of the device that created the alarm is passed in the alarm notification.

A notification that an alarm has been cleared or updated is sent to an application when SANM receives such a notification from SPECTRUM™, but only if the alarm which is being cleared or updated was initially sent to the application when it occurred (i.e., it passed
25 the filter criteria for that application).

2.0 Configuration Tool

The SANM Configuration Tool enables the user to define Alarm Notification Policies and to associate these policies with the applications that use SANM.

30 The Configuration Tool is invoked by selecting Notification Manager from the asterisk menu of SpectroGRAPH™.

2.1 Associations Window

When the Configuration Tool is invoked, the first window to appear is the Associations window, shown in Fig. 4. This window displays a list of the currently defined SANM applications and the policy that is associated with each of them.

5 A new association is created by selecting New from the File menu. This brings up the New Association window shown in Fig. 5.

An existing association is modified by selecting the association and then selecting Modify from the File menu. This brings up the Modify Association window shown in Fig. 6.

10 An existing association is deleted by selecting the association and then selecting Delete from the File menu. The selected association is deleted after the user confirms the operation in a Confirmation Dialog window (not shown).

The modification of an existing association can be scheduled by selecting the association and then selecting Schedule from the File menu. This brings up the Scheduler
15 window shown in Fig. 7.

All currently defined policies can be viewed by selecting Policies from the Tools menu. This brings up the Policies window shown in Fig. 8.

2.2 New Association Window

20 The New Association Window is illustrated in Fig. 5. In this window, a policy is selected from the list of available policies and the application name is entered. When OK is pressed, the window disappears and the new association appears in the Associations window (Fig. 4).

2.3 Modify Association Window

25 The Modify Association window is illustrated in Fig. 6. In this window, the user picks a policy from the list of available policies to associate with the selected application (SpectroPHONE™ in this example, available from Cabletron Systems, Inc.). Pressing OK makes this window disappear and the modified association is displayed in the
30 Associations window (Fig. 4).

2.4 Scheduler Window

The Scheduler window is illustrated in Fig. 7. Pressing the Associate button brings up the Modify Association window illustrated in Fig. 6. In the Modify Association window, the user picks a policy from the list of available policies to associate with the selected application (SpectroPHONE™ in this example). In the Scheduler window, the user then presses the Frequency button to specify the frequency of the association. The Frequency options are: Once, Hourly, Daily, Weekly and Monthly. The information in the area below the Frequency button changes depending on what frequency option is selected as follows:

- The Once option allows the user to specify the month, day and start-time.
- The Hourly option allows the user to specify the number of minutes after each hour.
- The Daily option allows the user to specify the time.
- The Weekly option allows the user to specify the day of the week and the time.
- The Monthly option allows the user to specify the day of the month and the time.

Once the desired scheduling options have been selected, pressing the Add button inserts the scheduling information into the Scheduled Entries portion of the window. Further entries can be added by repeating the previous steps. Entries can be modified and removed by selecting them and using the Modify and Remove buttons.

2.5 Policies Window

The Policies Window is illustrated in Fig. 8. This window shows all currently defined policies.

A new policy is created by selecting New from the File menu. This causes the New Policy window (Fig. 12) to appear.

An existing policy is viewed and modified by selecting the policy and then selecting Open from the File menu. This causes the Open Policy window (Fig. 9) to appear.

An existing policy is deleted by selecting the policy and then selecting Delete from the File menu. The selected policy is deleted after the user confirms the operation in a Confirmation Dialog window (not shown).

2.6 Open Policy Window

The Open Policy window is illustrated in Fig. 9. This window shows all the filters that make up the policy. In the example shown in Fig. 9, Filters 1 and 2 are visible, but subsequent filters can be viewed using the scroll bar on the right of the window. Similarly, the other filter parameters for Filter 1 and their associated values can be viewed using the scroll bar below the Filter 1 filter parameters.

To modify the displayed policy, Edit must be selected from the File menu. The View item in the menu bar then becomes Edit. Once in Edit mode, multiple values for a particular filter parameter can be deleted or negated by selecting the values and pressing the Delete or Negate button. Values can be added for a particular filter parameter by pressing the filter parameter button (e.g. Landscape or Model Type). This brings up a separate window containing a list of available values from which multiple values can be selected. An example of this window is shown in Fig. 10.

Filter parameters may be added to a filter by pressing the Parameter button within the filter. A pop-up menu appears containing all eight filter parameters. However, those filter parameters which are already present in the filter are greyed-out and cannot be selected. Selecting one of the available filter parameters from this menu causes the new filter parameter and associated value box to appear in the filter.

The alarm age for a particular filter can be modified by pressing the Age button in the Open Policy window. This brings up the Alarm Age window shown in Fig. 11. The values for the Hours and Minutes fields initially contain the values from the Age text field in the Open Policy window. These values can be modified using the up and down arrow buttons for hours and minutes.

A filter tag can be modified in the Open Policy window by typing directly into the Tag text field of a filter.

A new filter may be added to the policy displayed in the Open Policy window by pressing the Create Filter button. This will cause a new filter with no filter parameters to be added to the end of the list of filters.

An existing filter may also be duplicated. To do this the filter to be duplicated must first be selected by clicking within the filter label field (e.g. the area around the label Filter 2) and then pressing the Duplicate Filter button. Doing this causes a new filter, containing the same filter parameters and values as the selected filter, to be added to the end of the filter list. This new filter can then be modified.

After modifying a policy, Save can be selected from the File menu to save the modified policy under its existing name, or Save As can be selected to save the modified policy under a different name.

The information in the Open Policy window can be printed by selecting Print from the File menu.

2.7 New Policy Window

The New Policy Window is illustrated in Fig. 12. The operations that can be performed in the New Policy window are the same as those performed in the Open Policy window (Fig. 9). No filter parameters initially appear within Filter 1; therefore the first operation that needs to be performed is to select a filter parameter by pressing the Parameter button within Filter 1. All filter parameters are available from the pop-up menu at this point because the filter does not yet contain any filter parameters.

A new policy is saved by selecting Save As from the File menu and entering the name for the policy in a dialog box.

3.0 Integration of SANM and Application

A developer would use the following interface to integrate an application written in C or C++ with the Spectrum™ alarm mechanism.

An application using SANM to receive alarm notifications and to clear/acknowledge alarms requires two separate processes, as illustrated in Fig. 13.

As an example of how these two separate processes would be used in an application, the ARS Gateway™ product would use Process 1 to receive filtered alarms from SANM, format them into Trouble Tickets and put them into the ARS Database. Process 2 would be used when a user viewing a particular Trouble Ticket pressed a clear or acknowledge button in the Trouble Ticket.

Two different programming paradigms are required for the two application processes that use SANM:

For the process that receives alarm notifications from SANM, an asynchronous callback paradigm is used. This means that when the application code registers with SANM to receive alarms, it hands program control over to SANM. When SANM needs to send an alarm notification to the application, the application receives a callback from SANM. This process is terminated by sending it a TERM (terminate, 15) signal.

For the process that clears or acknowledges alarms, however, a synchronous paradigm is used. This means that the application code in this process has program control. When this application code makes a call to the SANM API to clear or acknowledge an alarm, the call blocks the application until it is finished.

3.1 Definitions and Data Structures

All definitions and data structures are contained in the SANM header file sanm.h and are described below.

The prototype for the application's callback functions is defined as follows:

```
typedef void (*SANM_Cb) (struct SANM_Alarm_Notify *);
```

All the data in an alarm notification is contained in the SANM_Alarm_Notify structure, which is defined as follows:

```
struct SANM_Alarm_Notify{
    char                *model_name;
    SANMULong           model_handle;
    char                *model_type_name;
    SANMULong           model_type_handle;
    int                 condition_value;
    char                *security_string;
    SANMULong           alarm_ID;
    SANMTimestamp       alarm_time;
```

- 19 -

```

SANMUInt long    cause_code;
char             *probable_cause;
char             *alarm_status;
char             *event_message;
5 char           *repair_person;
char             *IP_address;
char             *location;
SANMUInt long    severity;
SANMUInt long    alarm_age;
10 char          *SpectroSERVER_host;
char             *landscape;
SANMBoolean      user_clearable;
char             *filter_tag;

```

};

15 All errors and warnings are defined in the enumeration SANM_error as follows:

```
enum SANM_error
```

{

```

20 SANM_RETURN_OK,
SANM_INVALID_ALARM,
SANM_INVALID_LANDSCAPE,
SANM_ALARM_NOT_CLEARABLE,
SANM_REGISTER_ERROR

```

25 }

3.2 Functions

The functions that make up the SANM C/C++ API are described in the following sections in manual page format.

- 20 -

3.2.1 SANMInit

NAME

SANMInit - initialize interaction with SANM

SYNOPSIS

#include "sanm.h"

SANM_error SANMInit (char *application_name,
SANMBoolean rcv_or_clr);

DESCRIPTION

SANMInit serves to initialize the program for interaction with SANM. This function should be called from within both application processes before any other function in the SANM API.

INPUT ARGUMENTS

application_name

the name which must be used by the user to identify this application when using the Configuration Tool to associate a policy with it.

rcv_or_clr

a flag which indicates whether this process is going to receive alarm notifications or clear/acknowledge alarms. The flag can take either of the following two values:

SANM_RCV_ALARMS

SANM_CLR_ALARMS

RETURN VALUES

status

The return value will be one of the following values:

SANM_RETURN_OK

3.2.2 SANMRegister

NAME

SANMRegister - register with SANM

- 21 -

SYNOPSIS

#include "sanm.h"

5 SANM_error SANMRegister (SANMCb set_cb,
 SANMCb clear_cb,
 SANMCb update_cb);

DESCRIPTION

10 SANMRegister registers the application to receive
alarm notifications from SANM. By calling this
function, the application hands program control
over to SANM until one of the application's
callback functions is called.

15 INPUT ARGUMENTS

set_cb the name of the function that SANM will call
in order to send an alarm notification for a new
alarm. All applications must pass a valid
function for this parameter.

20 clear_cb the name of the function that SANM will call
in order to send an alarm notification for a
cleared alarm. This parameter can be NULL if
the application does not want to receive
25 notifications for cleared alarms.

30 update_cb the name of the function that SANM will call
in order to send an alarm notification for an
updated alarm. This parameter can be NULL
if the application does not want to receive
notifications for updated alarms.

- 22 -

RETURN VALUES

status

In normal operation, this function will never return. However, if it fails, one of the following errors will be returned:

SANM_REGISTER_ERROR

3.2.3 SANMClear

NAME

SANMClear - clear an alarm

SYNOPSIS

#include "sanm.h"

```
SANM_error SANMClear ( SANMULong alarm_ID,  
                      char *landscape );
```

DESCRIPTION

SANMClear clears an alarm in SPECTRUM. An application can only clear alarms for which it received notifications from SANM. Also, the user_clearable flag must have been set to CLEARABLE in the alarm notification

INPUT ARGUMENTS

alarm-ID

the ID of the alarm to be cleared

landscape

the landscape that generated the alarm

RETURN VALUES

status

The return value will be one of the following values:

SANM_RETURN_OK

SANM_INVALID_ALARM

SANM_INVALID_LANDSCAPE

- 23 -

SANM_ALARM_NOT_CLEARABLE

3.2.4 SANMAck

NAME

SANMAck - acknowledge an alarm

SYNOPSIS

#include "sanm.h"

SANM_error SANMAck (SANMUlong alarm_ID,
char *landscape);

DESCRIPTION

SANMAck acknowledges an alarm in SPECTRUM. An application can only acknowledge alarms for which it received notifications from SANM.

INPUT ARGUMENTS

alarm_ID

the ID of the alarm to be acknowledged

landscape

the landscape that generated the alarm

RETURN VALUES

status

The return value will be one of the following values:

SANM_RETURN_OK

SANM_INVALID_ALARM

SANM_INVALID_LANDSCAPE

The present embodiments may be implemented in a general purpose computer 70 as shown in Fig. 14. The general purpose computer may include a computer processing unit (CPU) 71, memory 72, a processing bus 73 by which the CPU can access the memory, and interface 74 to the rest of the alarm notification manager.

In alternative embodiments, the invention may be a computer apparatus which performs the functions of any of the previous embodiments. Alternatively, the

invention may be a memory, such as a floppy disk, compact disk, or hard drive, that contains the computer program or data structure, for providing to a general purpose computer instructions and data for carrying out the functions of the previous embodiment.

5 SpectroPHONE™, an Application that Incorporates SANM

In an alternative embodiment of the present invention, a telephonic alarm notification method and apparatus incorporates the capabilities of the SANM. This enables one telephonic alarm method and apparatus to handle alarms from multiple segments in a large and complex communications network.

10 Fig. 16 shows a block diagram illustrating this embodiment of the present invention. A telephonic alarm notification method and apparatus 186 comprises an alarm monitor 178 and a notification manager 184. A policy administrator 170 (such as that of SANM in the prior embodiment) is used to create a policy model 172 which is sent to the alarm monitor 178. Alarms arising from multiple segments of a communications network,
15 such as a first network segment 174 and a second network segment 176, are sent to the alarm monitor 178. These segments also send information regarding the users to be notified for each alarm.

The alarm monitor 178 determines which alarms pass the criteria specified by the policy model 172, and information regarding those alarms (the critical alarms) that
20 pass the criteria are put into two external files for use by the notification manager 184. An alarm information file 180 contains information regarding the critical alarms and the user information file 182 contains a list of respective users to be notified for each critical alarm.

Typically, one policy in the policy model 172 is in effect for the notification method at any given time. However, this policy can be changed by a user via the policy
25 administrator 170 (as in the SANM of the prior embodiment). All alarms, regardless of the network segment origin, are filtered through the filter of the policy model 172.

The notification manager 184 periodically reads the alarm information file 180 and initiates notifications based on the information contained in both files 180 and 182. For a critical alarm in file 180, the corresponding user(s) in file 182 are automatically
30 notified via a public telephone communications system 194. That user may be notified via a telephone 188, a pager 190, or another type of device 192 on the public telephone communications system 194.

- 25 -

Alternatively, a user can call into the notification manager 184 via the public telephone communications system 194. In that case, the notification manager 184 reads the information in the two files 180 and 182, and notifies the calling user of critical alarms that list the calling user as a user to be notified.

5 Once a user (such as a system administrator) is notified of a critical alarm, the user can take remedial actions to correct the problems associated with the critical alarms in the communications network. The user can then clear the critical alarm from the alarm information file 180 by calling into the notification manager 184. If a critical alarm is cleared, the information associated with that critical alarm is cleared from both files 180
10 and 182.

As alarms occur or are cleared on a communications network, they are handled via unsolicited request management which means that as soon as alarms are detected and filtered or are cleared, their associated information is adjusted in the two files 180 and 182 in real-time. These adjustments can be made by the alarm monitor 178 while
15 the notification manager 184 is performing other activities such as calling users or polling hardware on the public telephone communications system 194.

This division of labor in the telephonic alarm notification method 186 into the alarm monitor 178 and the notification manager 184 and a tight integration of these two portions allows for monitoring of critical alarms with high accuracy and timely
20 response by the users. In addition, the incorporation of the SANM capabilities into the telephonic alarm notification method allows for sophisticated filtering of the alarms via the policy administrator 170 and for monitoring of alarms from multiple network segment servers.

Fig. 17 shows a typical system architecture for the new SpectroPHONE™
25 application, which is an example embodiment of the telephonic alarm notification method and apparatus of the present invention. A host machine 200 runs the new SpectroPHONE™ application which incorporates the SANM 204 of the prior embodiments.

A user can specify policy models through the SpectroPHONE™ Graphical
30 User Interface (GUI) 206. The policy administrator 208 sends the prespecified policy models to multiple distributed network managers 210 and 214 (in remote machines 212 and 216 respectively) that serve separate network segments. The distributed network

managers 210 and 214 send information regarding alarms generated in their respective network segments and regarding policies to the SANM 204 and information regarding users to be notified of those alarms to SpectroPHONE™ 202. SpectroPHONE™ 202 then notifies users when critical alarms arise via the Computerfone Hardware 218 which is the intermediary to the external public communications medium 220.

The new SpectroPHONE™ application also provides additional user administration functions via the SpectroPHONE™ GUI 206. Fig. 18 shows an example GUI window display 230 having a SpectroPHONE™ attributes field 232. In this field, the user can specify a predetermined location number on the public telephone communications system SpectroPHONE™ should call for notification of critical alarms corresponding to that user. This location number can be for a telephone or pager.

In addition, the user can enter a password corresponding to that user in the SpectroPHONE™ attributes field. SpectroPHONE™ will ask a called user to verify the called user's identification by entering the password before that called user can be notified of the critical alarms. Thus, the new SpectroPHONE™ application provides added security for access to alarm information.

The user can also enter retry time intervals in the SpectroPHONE™ attributes field 232. SpectroPHONE™ will automatically retry calling the user every time interval if prior calls to the user have failed.

Finally, a tag-field in the SANM 204 incorporated into the SpectroPHONE™ application 202 is used to provide an escalation function. In this function, a user can specify a chain of users to be notified when alarms pass a given policy-based filter. Thus, an entry in the tag-field is associated with a given policy-based filter. If a prior user in the chain cannot be reached, a subsequent user in the chain is called until a user in the chain can be reached or until the last user in the chain cannot be reached.

The GUI 206 and the tag-field in the SANM provide more user-friendly functions in SpectroPHONE™. These features provide the communications network administrator with more control over notification regarding critical alarms that arise on the network, via the public telephone communications system.

Having thus described certain particular embodiments of the invention, various modifications will readily occur to those skilled in the art which are intended to be within the scope of this invention. Accordingly, the foregoing description is by way of example only, and not intended to be limiting.

CLAIMS

1. A method of alarm notification comprising the steps of:
 - (a) receiving alarms from multiple servers;
 - (b) assigning policy-based filters to one or more associated applications; and
 - (c) applying the assigned policy-based filters to the alarms and for the alarms which pass the filters, generating an alarm notification and forwarding the same to the one or more associated applications.
2. The method of claim 1, wherein:
the assigning step includes assigning a plurality of filters comprising a policy to the one or more associated applications.
3. The method of claim 2, wherein:
each filter comprises at least one filter parameter; and
the applying step comprises performing a logical AND of all parameters within one filter and performing a logical OR between all filters within one policy.
4. The method of claim 3, wherein:
the generating step includes specifying real-time values of each filter parameter in the alarm notification.
5. The method of claim 2, wherein:
the assigning step includes storing a policy name and the one or more associated applications in a database accessible to all servers.
6. The method of claim 1, wherein:
the assigning step includes assigning a tag to each filter.
7. The method of claim 6, wherein:
the generating step includes specifying the tag for the filter which the alarm passed in the alarm notification.

8. The method of claim 1, wherein:

the servers are network management servers and the one or more applications are network management applications.

5 9. The method of claim 1, wherein:

the generating step further includes specifying a user name in the alarm notification to enable the application which receives the alarm notification to notify a user having the specified user name.

10 10. The method of claim 1, wherein:

the assigning step includes scheduling the assigning to occur at a specified time.

11. The method of claim 1, further comprising:

15 (d) following resolution of an alarm, forwarding an alarm clear message to the associated application.

12. The method of claim 1, wherein:

the assigning step includes assigning the same filters to multiple associated applications.

20 13. The method of claim 1, wherein:

the assigning step is performed by a user via a graphical user interface.

14. The method of claim 1, wherein:

25 the generating step includes generating an alarm notification which contains information about the device which generated the alarm.

15. The method of claim 1, further comprising:

30 one or more of the applications generating an alarm clear message and forwarding the same to the server which generated the alarm.

- 30 -

16. Apparatus for alarm notification comprising:

a database of policy-based filters;

a user interface for assigning policy-based filters to one or more associated applications;

a processor and a memory device containing a program of instructions for the processor which instructions include:

means for receiving alarms from a plurality of servers;

means for applying policy-based filters to the alarms and generating an alarm notification for those alarms which pass the filters; and

means for forwarding the alarm notification to the one or more associated applications.

17. A method of notifying a user on a communications network of alarms generated on the communications network, the method comprising the steps of:

a) receiving alarms from multiple servers of the communications network;

b) applying a policy comprised of a plurality of filters to the alarms to determine critical alarms that pass the filters; and

c) notifying the user of the critical alarms via a public communications system.

18. The method of claim 17, further comprising the steps of:

the user sending a request message, via the public communications system, to be notified of the critical alarms; and

waiting for the request message from the user before the step of notifying.

19. The method of claim 17, further comprising the step of:

storing alarm information and user information associated with the critical alarms.

20. The method of claim 17, further comprising the step of:

the user determining the policy by selecting the plurality of filters that comprise the policy.

21. The method of claim 17, further comprising the step of:
the user setting the filter parameters of the plurality of filters.
22. The method of claim 17 wherein, the step of notifying includes the steps of:
the user specifying predetermined time intervals for automatic notification of the
critical alarms; and
automatically notifying the user at the predetermined time intervals.
23. The method of claim 17 wherein, the step of notifying includes the steps of:
the user specifying a predetermined location number on the public communications
system for receiving notification of the critical alarms; and
calling the predetermined location number for notification of the critical alarms.
24. The method of claim 17 wherein, the step of notifying further includes the steps of:
the user specifying a subsequent location number for receiving notification of the
critical alarms; and
if the predetermined location number cannot be reached, calling the subsequent
location number for notification of the critical alarms.
25. The method of claim 17 further comprising the steps of:
calling a called user for notification of the critical alarms;
asking the called user to enter a password when the called user answers; and
not notifying the called user of the critical alarms if the password is not a valid
password.
26. The method of claim 26, further comprising the step of:
the user setting the valid password that the called user verifies.
27. The method of claim 17, further comprising the steps of:
the user specifying a communications device on the public communications system;
and
notifying the user of the critical alarms via the communications device.

28. The method of claim 27 wherein, the communication device is one of a telephone and a pager.

29. An apparatus for notifying a user on a communications network of alarms
5 generated on the communications network, the apparatus comprising:
a policy administrator for creating a policy comprised of a plurality of filters;
an alarm monitor, operatively connected to the policy administrator, for receiving
alarms from multiple servers of the communications network and for applying the policy to
the alarms to determine critical alarms that pass the filters; and
10 a notification manager, operatively connected to the alarm monitor and a public
communications system, that notifies the user of the critical alarms via the public
communications system.

30. The apparatus of claim 29, further comprising:
15 a database, operatively connected to the alarm monitor and the notification
manager, for storing alarm information and user information associated with the critical
alarms.

31. The apparatus of claim 29, further comprising:
20 a user administrator that inputs from the user, a valid password that a called user
must verify before the called user is notified of the critical alarms.

32. The apparatus of claim 29 wherein, the notification manager waits for a request
message from the user before notifying the user of the critical alarms.

25 33. The apparatus of claim 29 wherein, the policy administrator inputs, from the user, a
selection of filters that define the policy.

34. The apparatus of claim 29 wherein, the policy administrator inputs, from the user,
30 filters parameters that define the plurality of filters.

- 33 -

35. The apparatus of claim 29, further comprising:

a graphical user interface that inputs, from the user, a predetermined location number on the public communications system, the notification manager calling the predetermined location number for notification of the critical alarms.

5

36. The apparatus of claim 35 wherein, the graphical user interface inputs, from the user, a subsequent location number to be called for notification of the critical alarms if the predetermined location number cannot be reached.

10 37. The apparatus of claim 35 wherein, the graphical user interface inputs, from the user, predetermined time intervals when the notification manager shall automatically call the user for notification of the critical alarms.

15 38. The apparatus of claim 35 wherein, the graphical user interface inputs from the user, predetermined time intervals when the notification manager shall retry calling the predetermined location number if prior calls to the predetermined location number have failed.

1/13

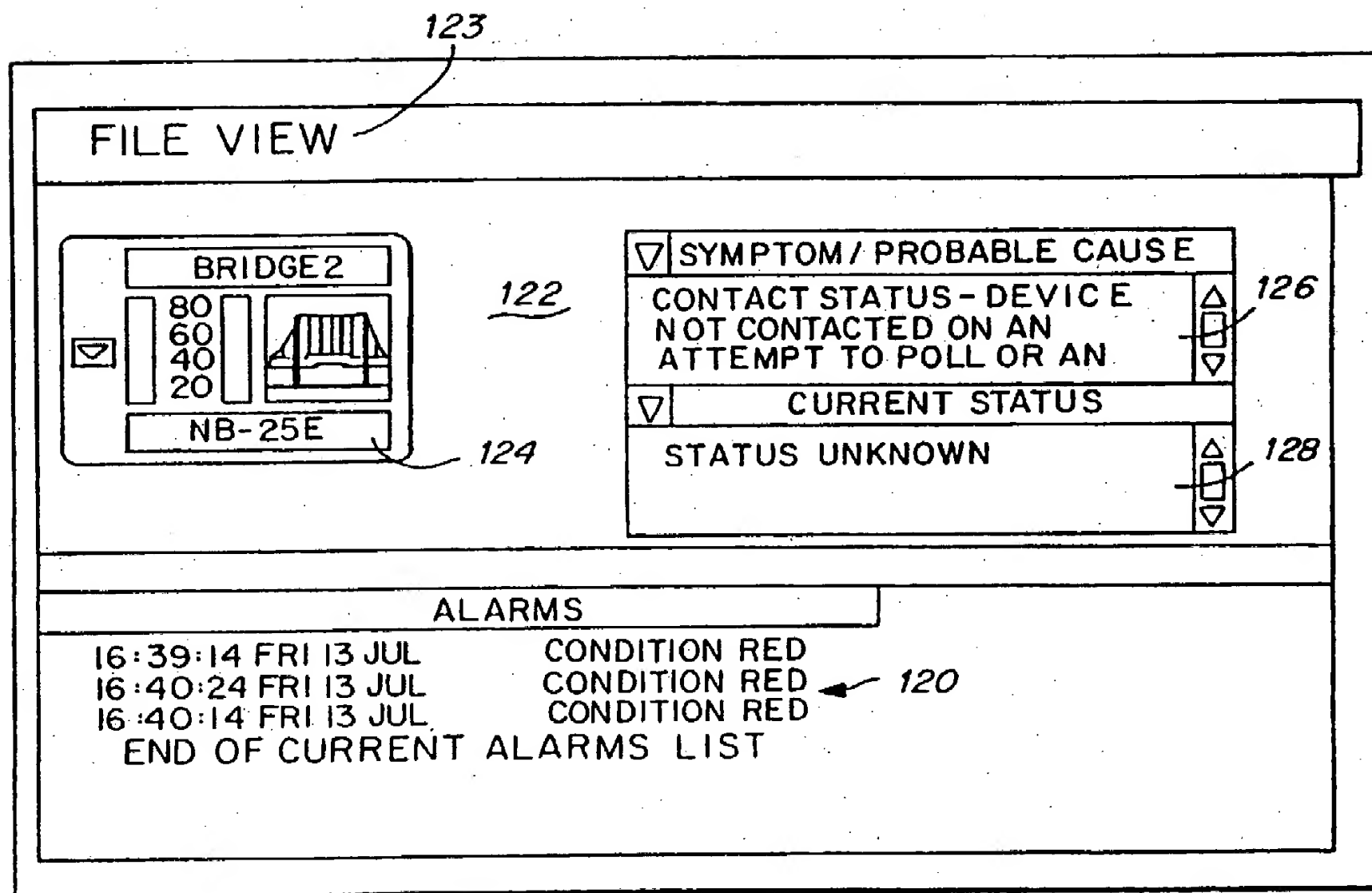


FIG. 1
(PRIOR ART)

2/13

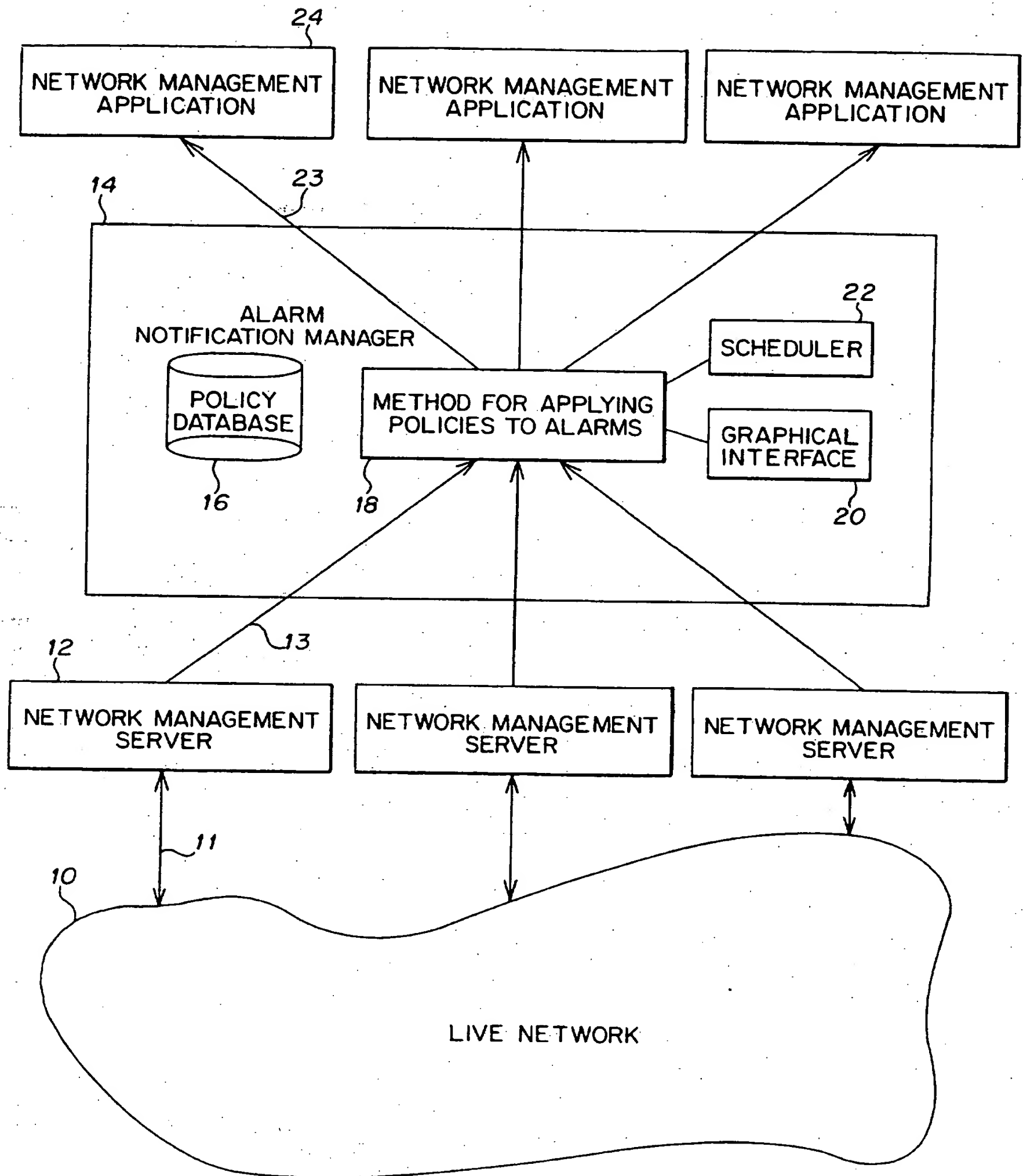


FIG. 2

3/13

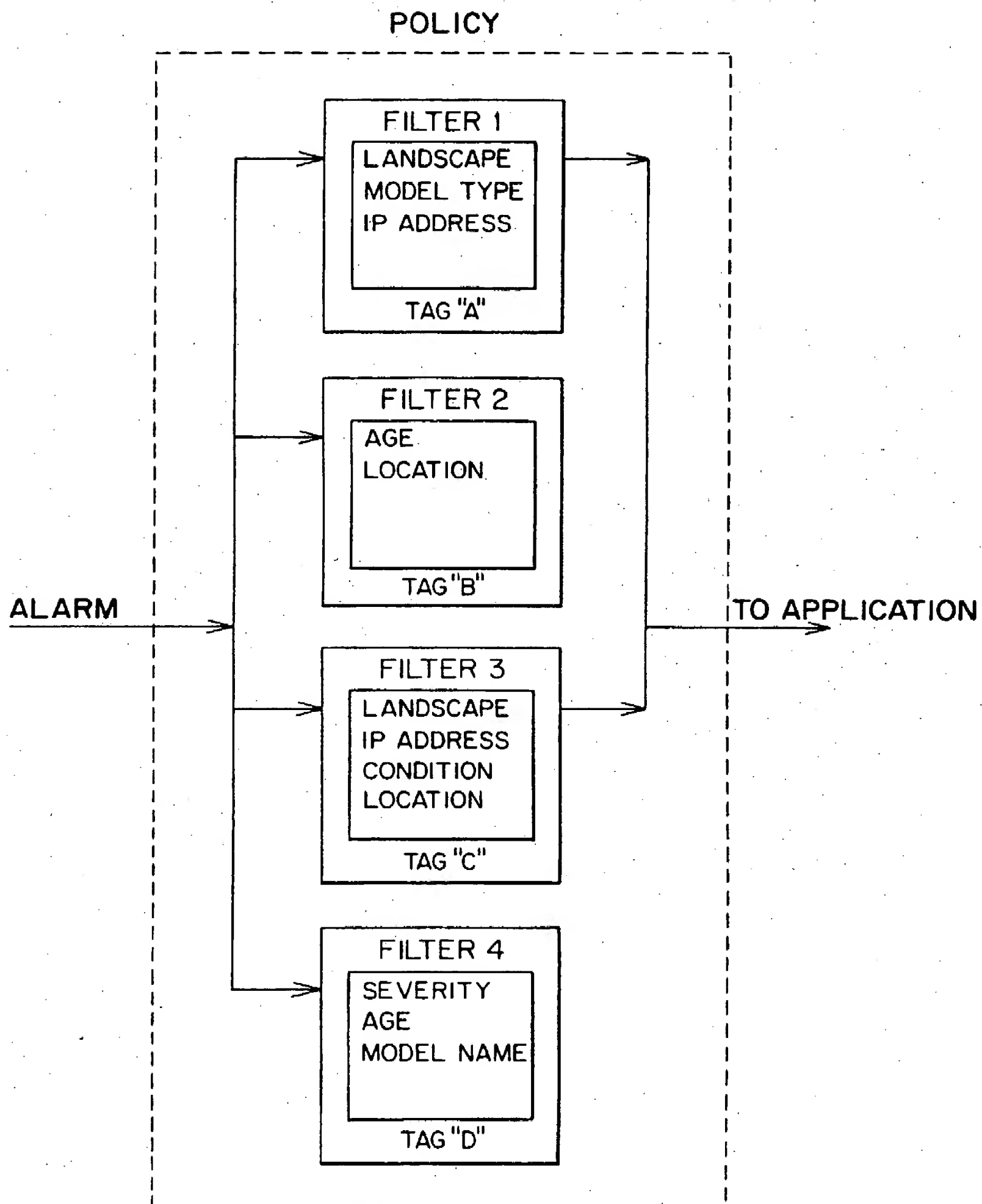


FIG. 3

4/13

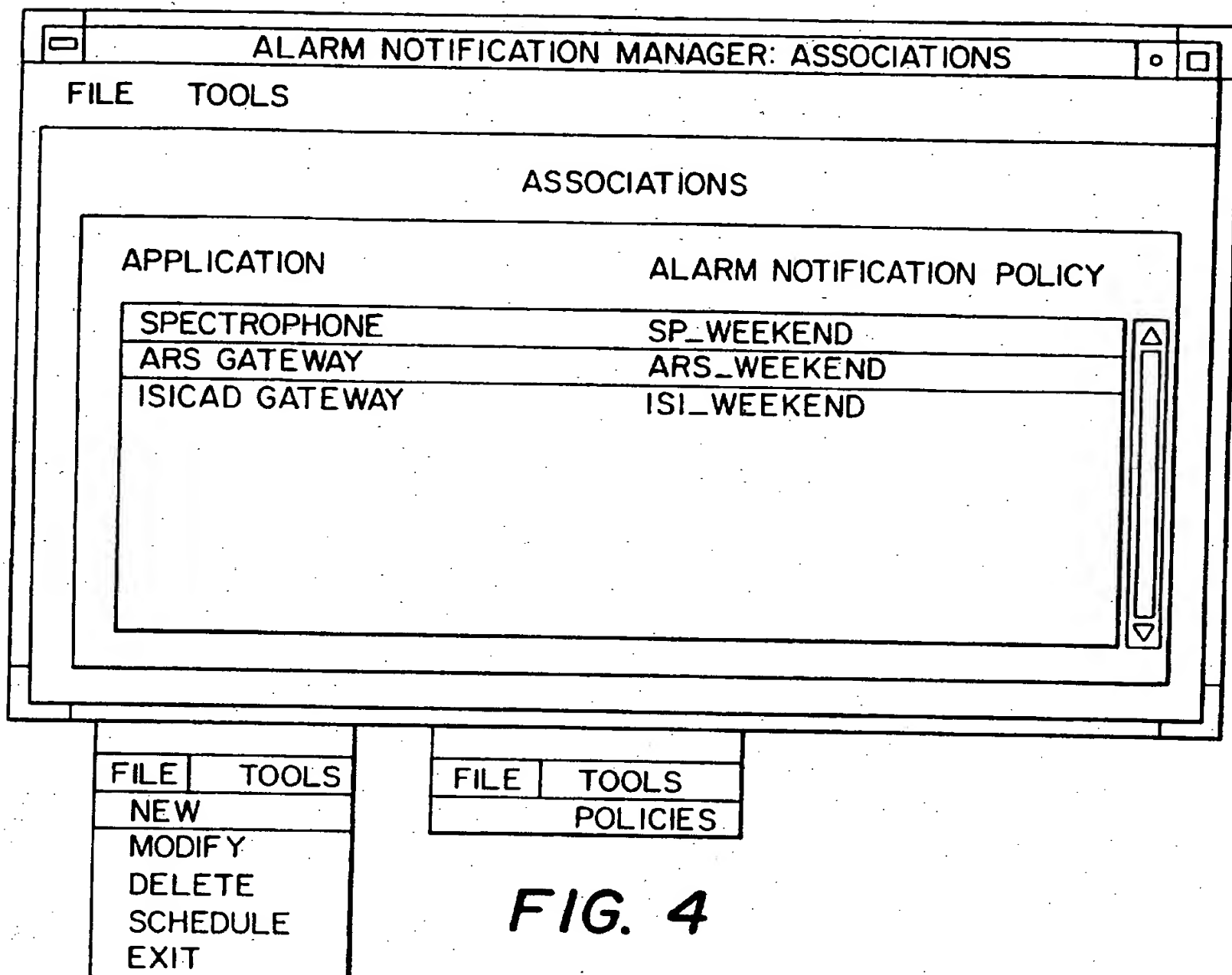


FIG. 4

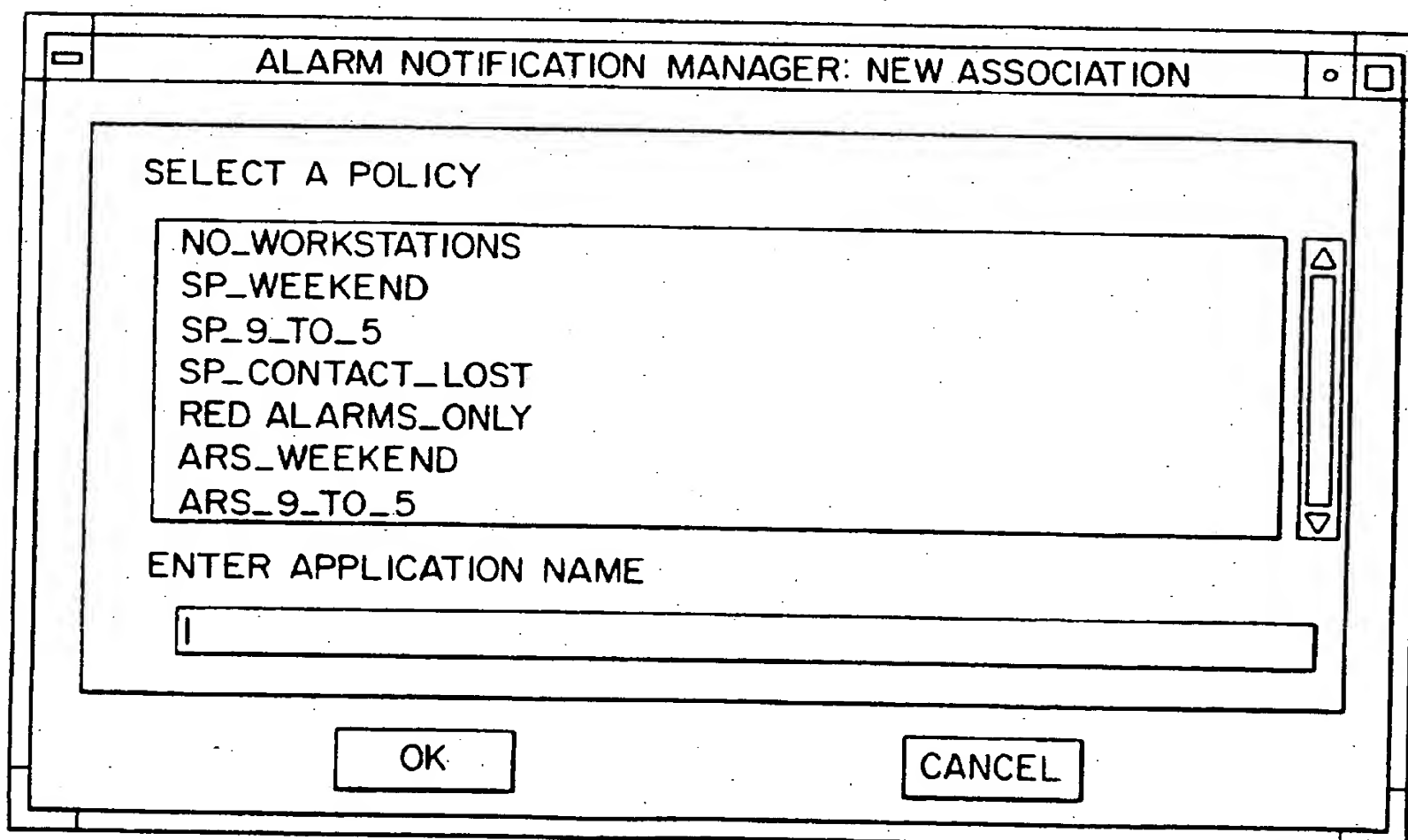


FIG. 5

5/13

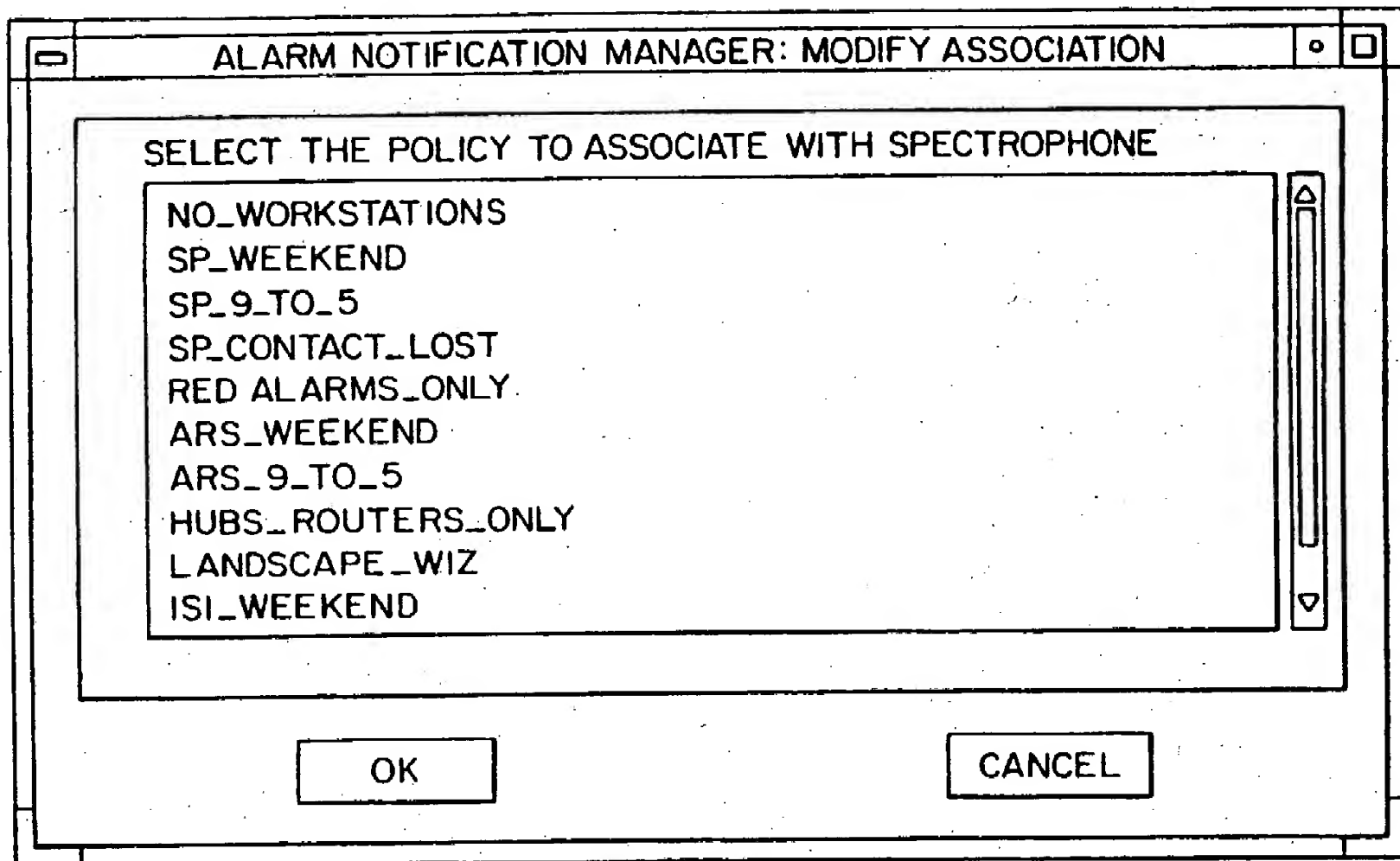


FIG. 6

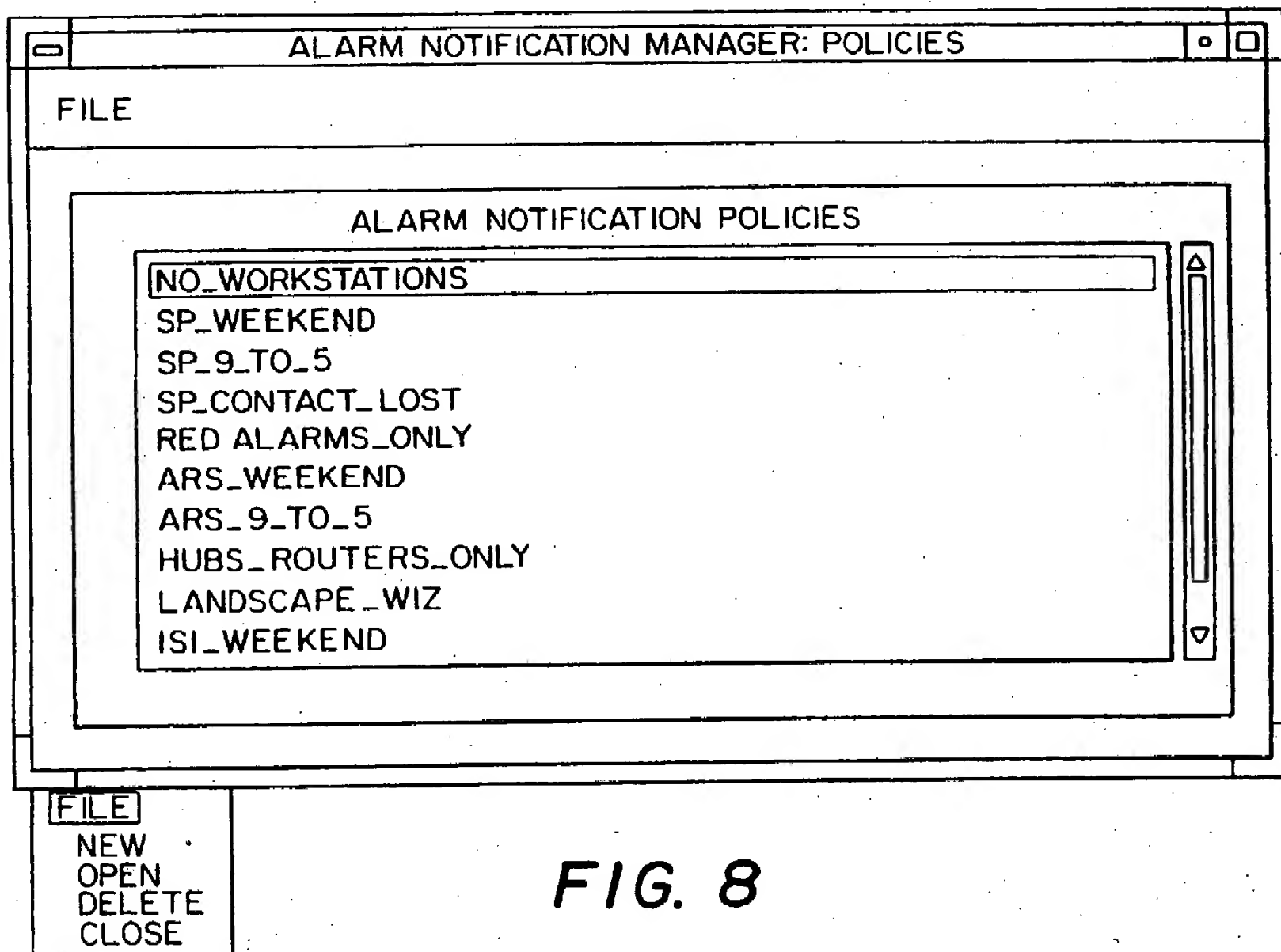


FIG. 8

6/13

SCHEDULER	
FEB 10	17:11
SCHEDULE A TASK	
<div>SCHEDULE <input type="button" value="ASSOCIATE"/></div> <div>COMMAND <input type="text"/></div> <div>FREQUENCY <input type="button" value="WEEKLY"/></div> <div style="margin-top: 10px;">REPEAT ON: <div style="display: flex; justify-content: space-around;"><div><input checked="" type="checkbox"/> SUN</div><div><input checked="" type="checkbox"/> MON</div><div><input checked="" type="checkbox"/> TUE</div><div><input checked="" type="checkbox"/> WED</div><div><input checked="" type="checkbox"/> THU</div><div><input checked="" type="checkbox"/> FRI</div><div><input checked="" type="checkbox"/> SAT</div></div><div style="margin-top: 10px;">AT <div><input type="text" value="Δ"/><input type="text" value="▽"/></div>:<div><input type="text" value="17"/><input type="text" value="00"/></div><div><input type="text" value="Δ"/><input type="text" value="▽"/></div></div></div>	<div style="text-align: center; margin-bottom: 10px;">SCHEDULED ENTRIES</div> <div style="border: 1px solid black; padding: 5px; min-height: 200px;">WEEKLY ON MON AT 08:00 "ASSOCIATE APP=SPECTROPHONE" WEEKLY ON FRI AT 17:00 "ASSOCIATE APP=SPECTROPHONE"</div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"><div><input type="button" value="ADD"/> <input type="button" value="MODIFY"/> <input type="button" value="REMOVE"/></div><div><input type="text" value="Δ"/></div></div>
<input type="button" value="OK"/>	<input type="button" value="APPLY"/> <input type="button" value="RESET"/> <input type="button" value="CANCEL"/>

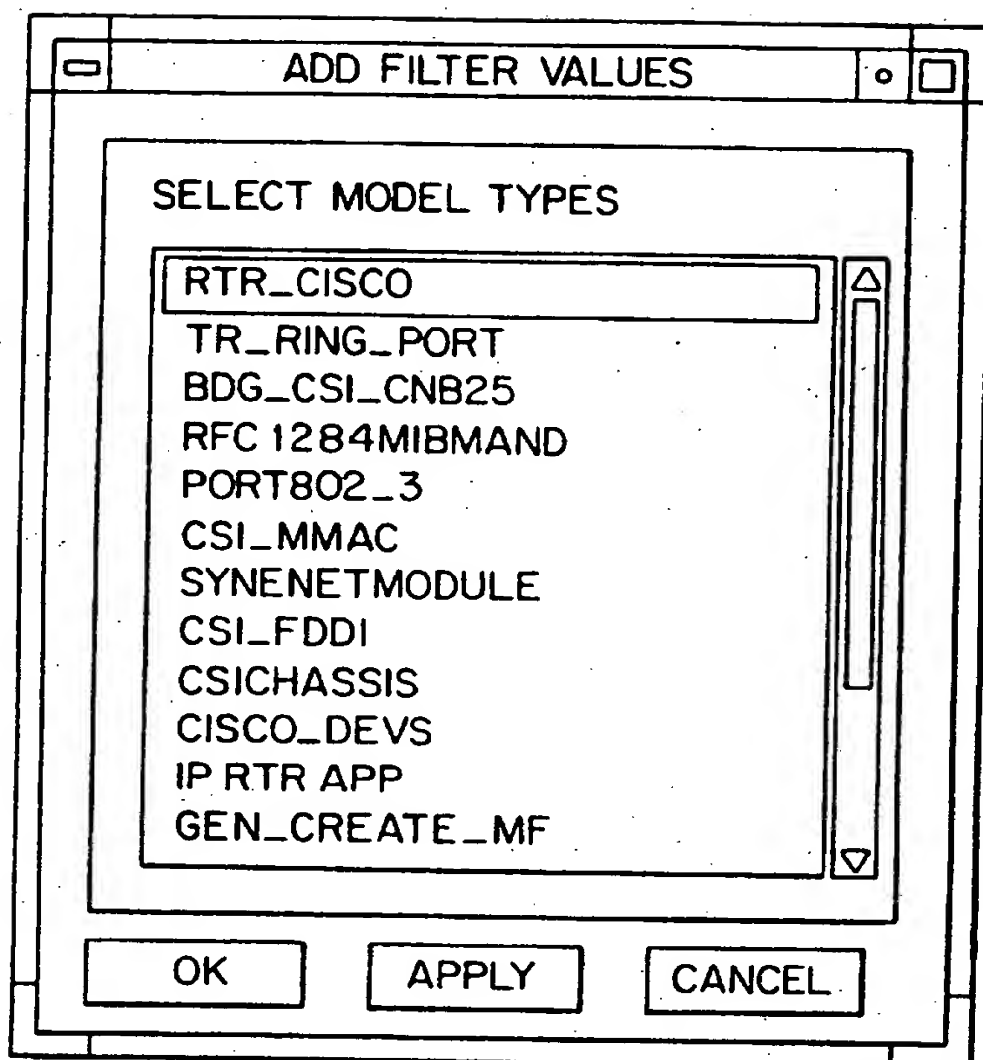
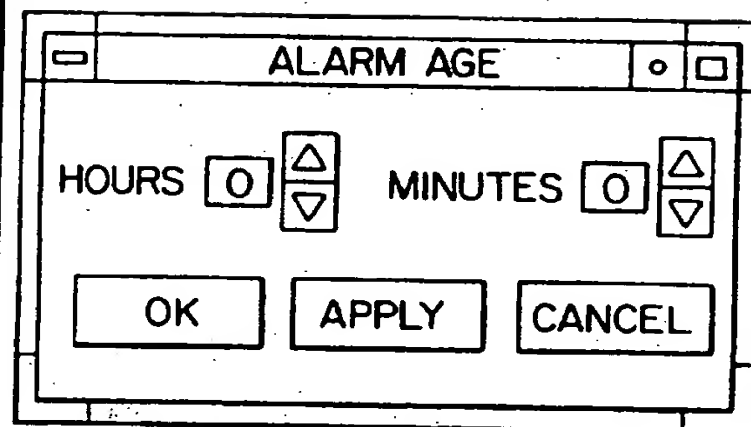
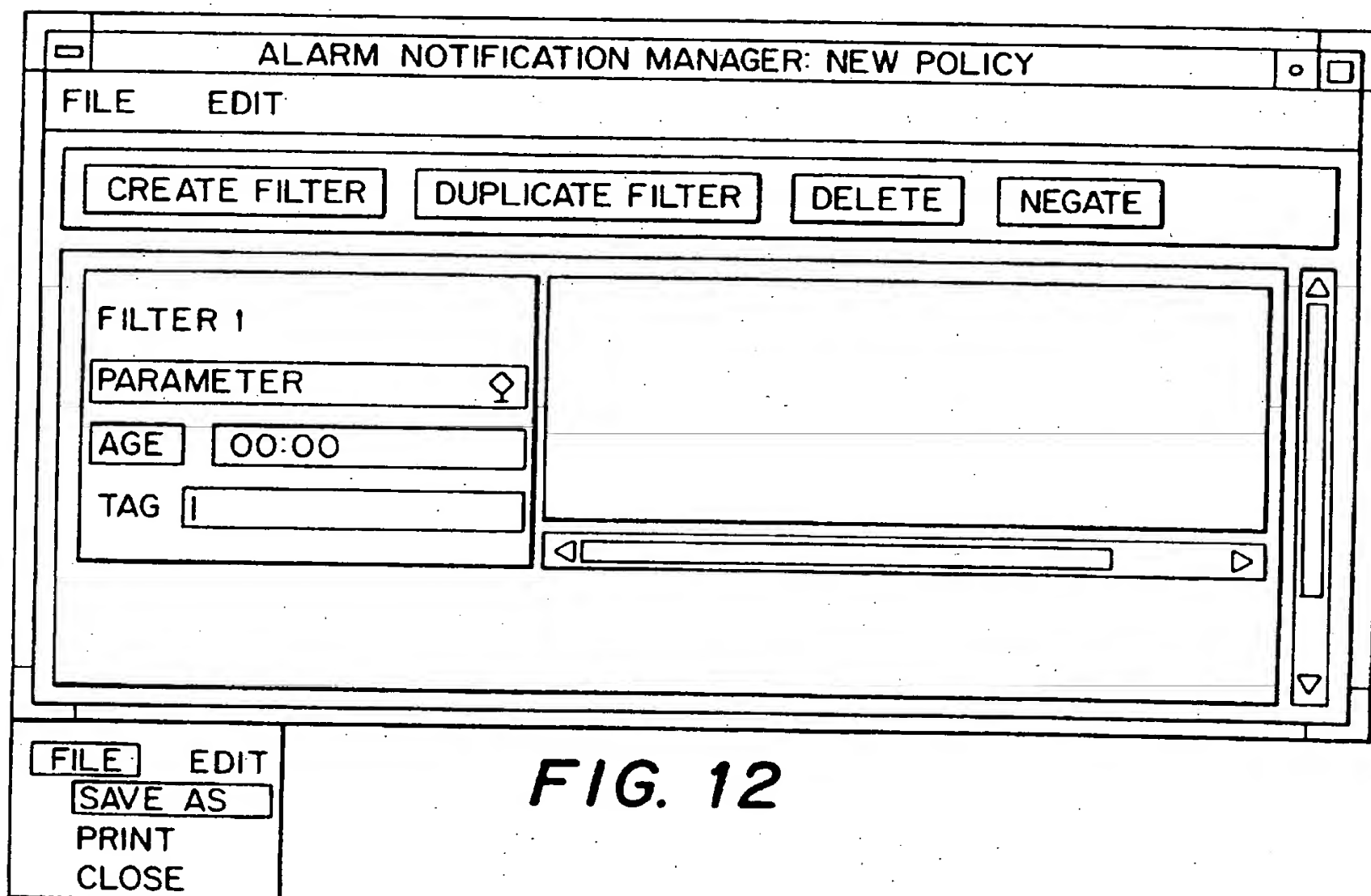
FIG. 7

7/13

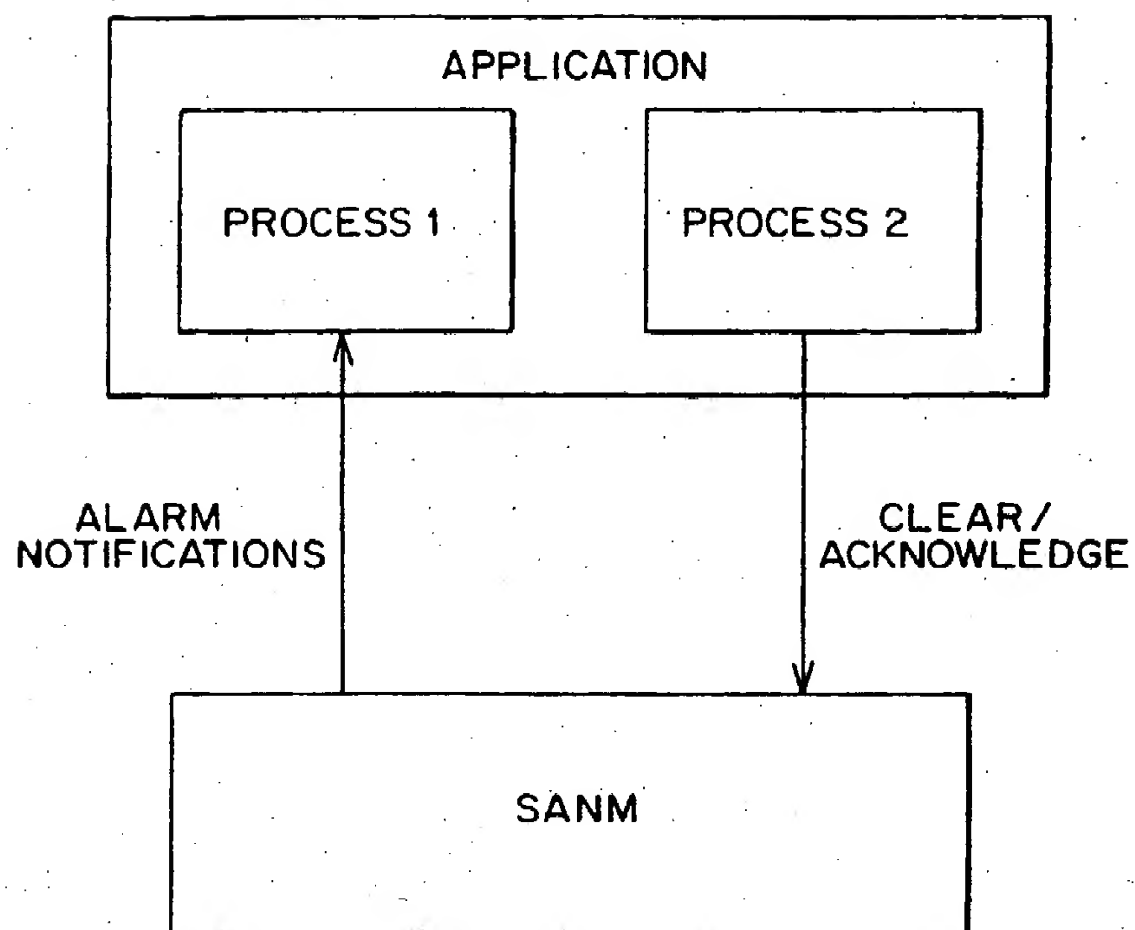
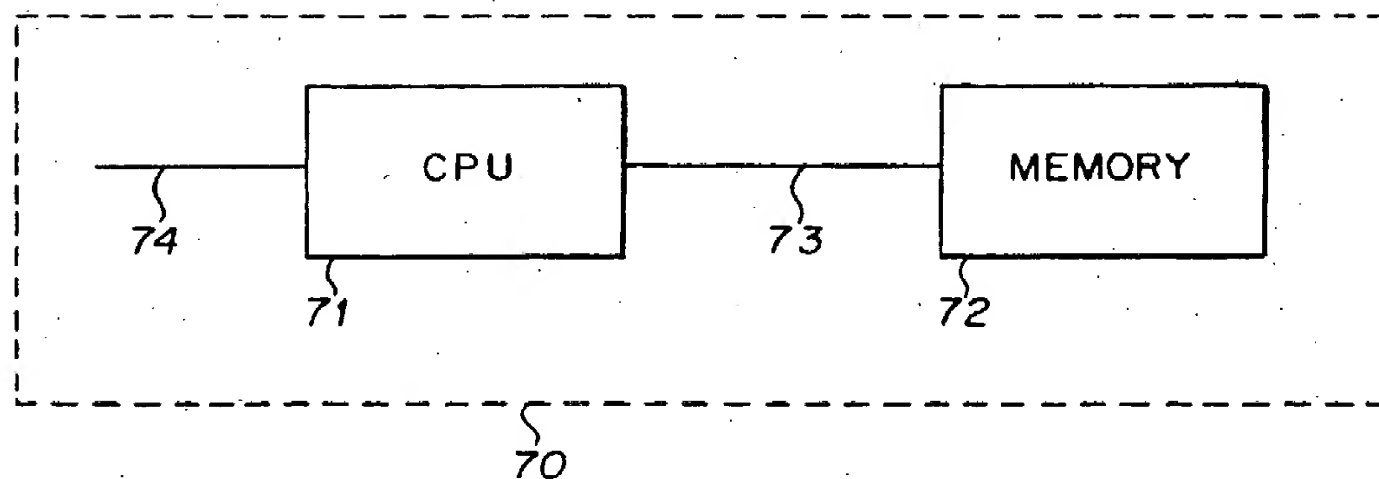
<div style="display: flex; justify-content: space-between;"> □ ALARM NOTIFICATION MANAGER: OPEN POLICY □ </div>	
<div style="display: flex; justify-content: space-between;"> FILE VIEW </div>	
<p>POLICY: SP_WEEKEND</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> CREATE FILTER DUPLICATE FILTER DELETE NEGATE </div>	
<p>FILTER 1</p> <div style="margin-bottom: 5px;"> PARAMETER ◇ </div> <div style="margin-bottom: 5px;"> AGE 02:30 </div> <div> TAG LEO LANFIXER </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> LANDSCAPE AND </div> <div style="border: 1px solid black; padding: 2px;"> WIZ BRAT ALTON HOUND CAPITAL </div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> MODEL TYPE AND </div> <div style="border: 1px solid black; padding: 2px;"> RTR_CISCO RTR_WELLFLEET RTR_IBM_WS RTR_PROTEON </div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> IP SUBNET AND </div> <div style="border: 1px solid black; padding: 2px;"> 134.141.141.0 134.141.143.0 134.141.65.0 </div> </div>
<p>OR</p>	
<p>FILTER 2</p> <div style="margin-bottom: 5px;"> PARAMETER ◇ </div> <div style="margin-bottom: 5px;"> AGE 00:05 </div> <div> TAG WENDY WANWORKER </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> MODEL NAME AND </div> <div style="border: 1px solid black; padding: 2px;"> SCMCISCO_1-2 SCMCISCO_2-2 134.141.67.194 MIB-11 </div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> SEVERITY AND </div> <div style="border: 1px solid black; padding: 2px;"> NOT YELLOW </div> </div>
<p>OR</p>	
<div style="display: flex; justify-content: space-between;"> FILE VIEW <div style="border: 1px solid black; padding: 2px;"> EDIT SAVE SAVE AS PRINT CLOSE </div> </div>	

FIG. 9

8/13

**FIG. 10****FIG. 11****FIG. 12**

9/13

**FIG. 13****FIG. 14**

10/13

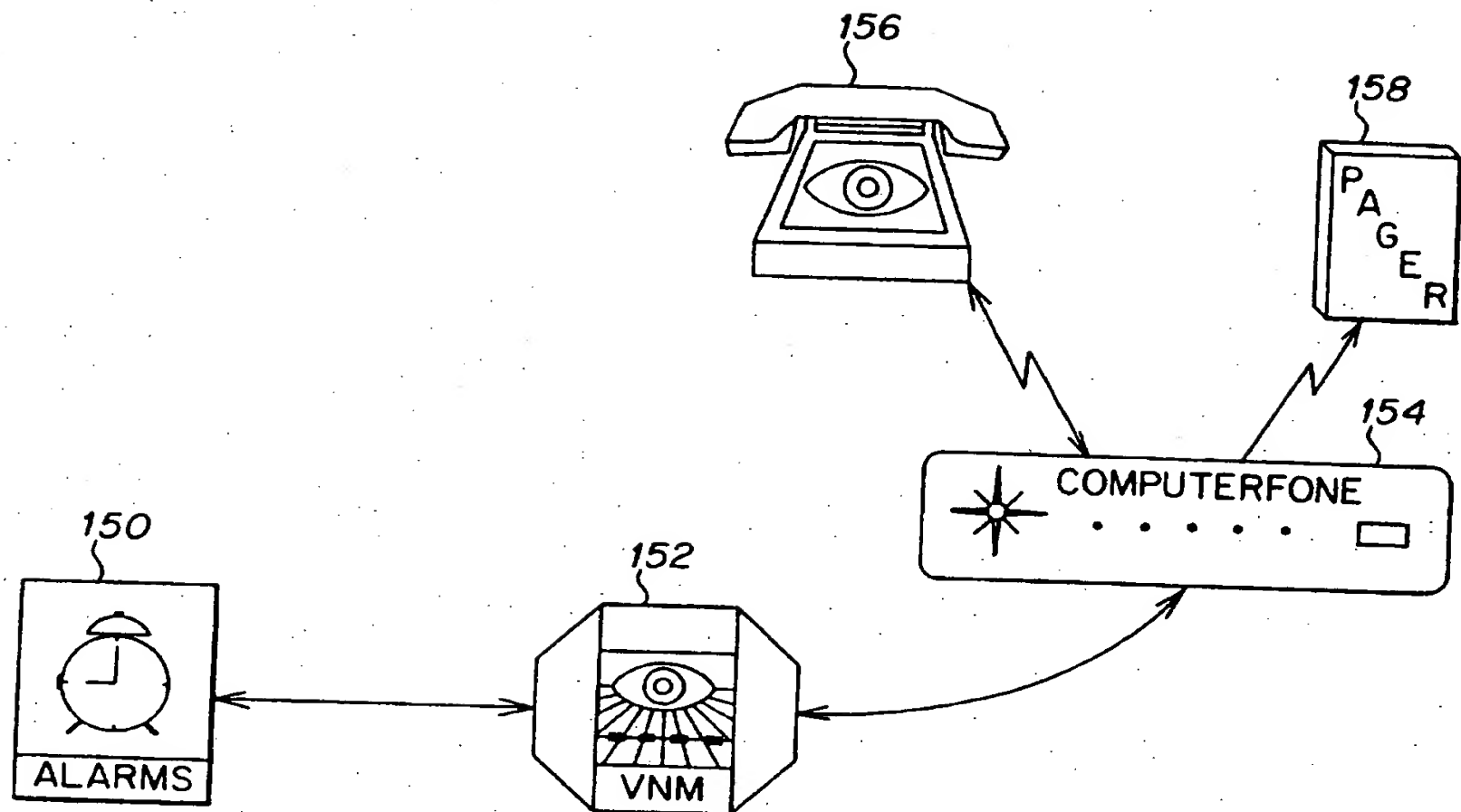


FIG. 15
(PRIOR ART)

11/13

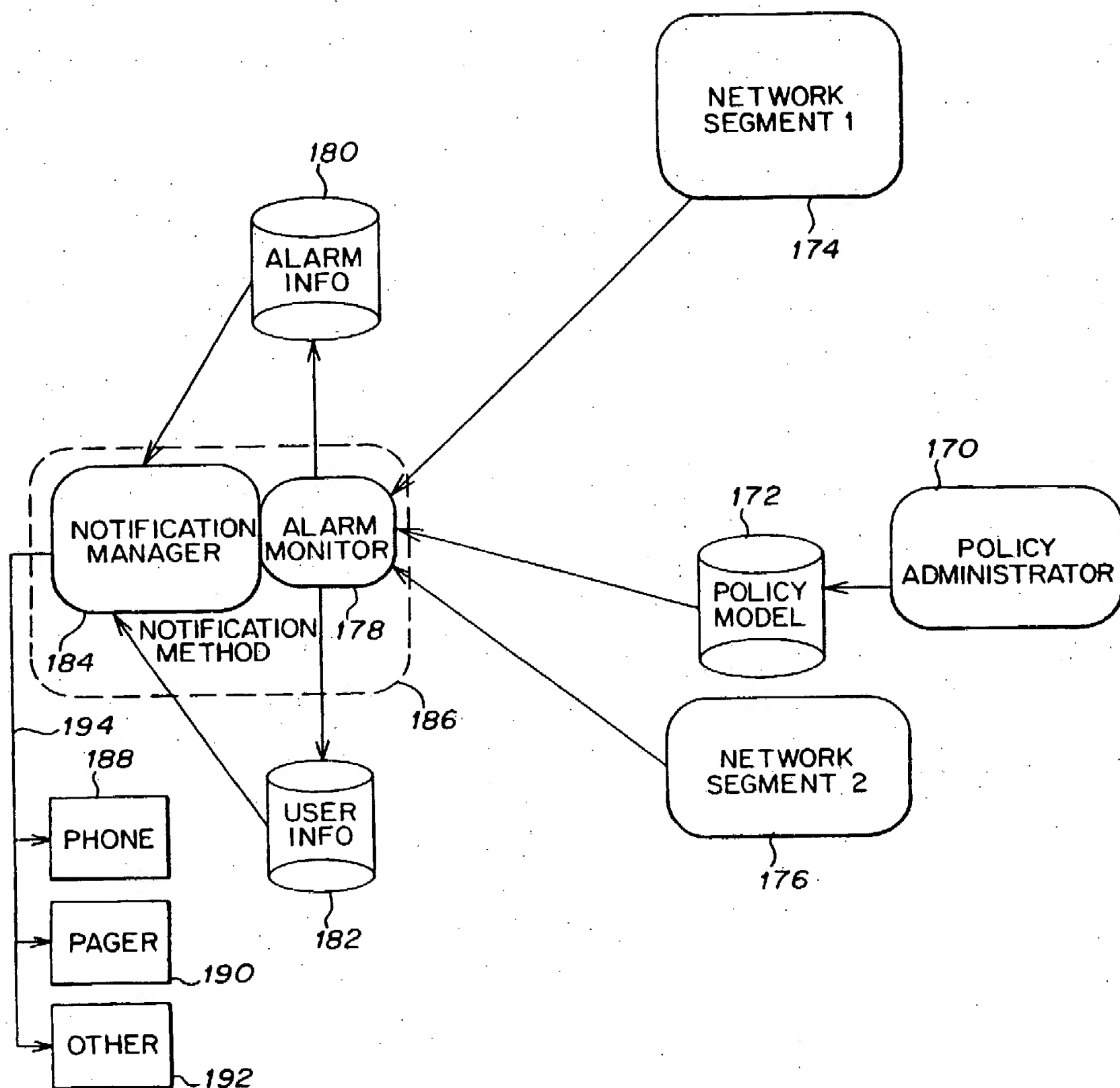


FIG. 16

12/13

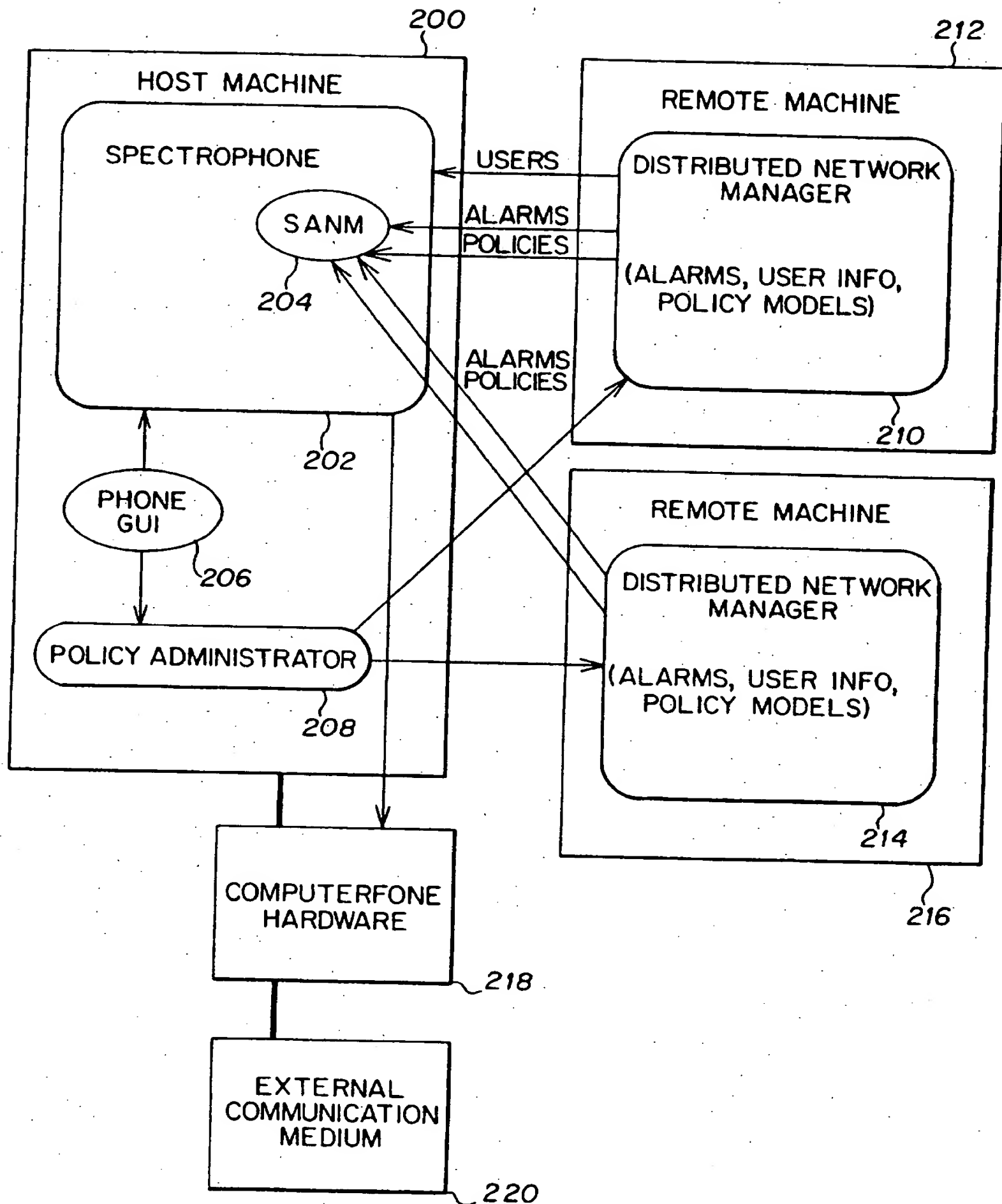


FIG. 17

13/13

230

PRIMARY LANDSCAPE 0x00640000-VNM VALDOR-LYNNB OF TYPE USER ◻

FILE VIEW HELP?

MODIFY USER

USER NAME	LYNNB
USER ID	
COMMUNITY STRING	ADMIN.O
USER SITE	
USER START	
USER DEPARTMENT	
USER FULL NAME	
USER STREET	
USER CITY	
USER STATE	
USER COUNTRY	
USER PICTURE FILE	

YOUR
PICTURE
HERE

DVNM USER STAT: HOME USER MODEL USER DOMAIN MAINTENANCE

SPECTROPHONE ATTRIBUTES

USER ID		PASSWORD	
NOTIFICATION TYPE	NONE	RETRY INTERVAL	10
PHONE NUMBER		RETRY COUNT	0
PAGER NUMBER		RETRY MAX	6
PAGER STRING		LAST NOTIFICATION	

232

FIG. 18

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 96/04332

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DATA COMMUNICATIONS, vol. 24, no. 1, 1 January 1995, pages 116-118, XP000480825 JANDER M: "REAL DISTRIBUTED MANAGEMENT" see page 116 - page 118 ---	1-38
X	DATA COMMUNICATIONS, vol. 24, no. 1, 1 January 1995, page 119/120 XP000480826 JANDER M: "ROUNDING OUT THE ROSTER OF SNMP AGENTS" see page 119 - page 120 --- -/-	1-4,8-15



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

& document member of the same patent family

Date of the actual completion of the international search

1 August 1996

Date of mailing of the international search report

16. 08. 96

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+ 31-70) 340-3016

Authorized officer

Goossens, A

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 96/04332

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PROCEEDINGS OF THE GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM), SAN FRANCISCO, NOV. 28 - DEC. 2, 1994, vol. 1 OF 3, 28 November 1994, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 548-552, XP000488606 KWANG-HUI LEE: "A DISTRIBUTED NETWORK MANAGEMENT SYSTEM" see page 549, column 2 - page 551, column 1	1,8
X	--- PROCEEDINGS OF THE MILITARY COMMUNICATIONS CONFERENCE (MILCOM), LONG BRANCH, NJ., OCT. 2 - 5, 1994, vol. 2 OF 3, 2 October 1994, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 644-649, XP000505952 SCHLAERTH J P: "A CONCEPT FOR TACTICAL WIDE-AREA NETWORK HUB MANAGEMENT" see page 647, column 1 - page 648, column 1 -----	1,8